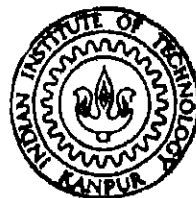# DESIGN AND IMPLEMENTATION OF A PC BASED PART PROGRAMMING PACKAGE FOR PLANAR MACHINING APPLICATIONS

by

K. KRITHIVASAN

INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

JANUARY, 1988

DESIGN AND IMPLEMENTATION OF A PC BASED PART PROGRAMMING PACKAGE FOR PLANAR MACHINING APPLICATIONS

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
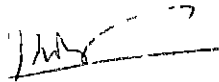
MASTER OF TECHNOLOGY

by

K. KRITHIVASAN

to the

INDUSTRIAL AND MANAGEMENT ENGINEERING PROGRAMME

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
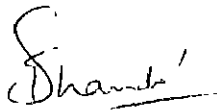
JANUARY, 1988

CERTIFICATE

It is certified that this work entitled 'Design and Implementation of a PC-based Part Programming Package for Planar MAchining applications' by K.Krithivasan has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Dr. J L.Batra
Professor
Dept of Industrial
and Management Engg.
I.I.T Kanpur

Dr.S.G.Dhande
Professor
Dept. of Mechanical Engg.
and Computer Science
I.I.T. Kanpur.

ACKNOWLEDGEMENTS

ABSTRACT

A PC-based graphic software for planar part programming applicatons has been designed. The system accepts the geometrical specifications graphically. Cutter motion specifications are provided by the user in a dialogue mode. The software has been tested successfully. The machining action has been simulated. It is applicable to wire elctro-discharge machining and flame cutting. The input required from the user is minimum. The software is written in GWBASIC. The whole system is menu driven. Some examples showing how the part programming system works are also provided.

CONTENTS

CHAPTER 1

INTRODUCTION

Numerical Control(NC) can be defined as a form of programmable automation in which the process is controlled by numbers, letters and symbols. Most of the developments in computerised design and manufacturing have a common origin in Numerical Control. The conceptual framework established during the development of NC is still undergoing refinement and enhancement with developments in computer hardware. In NC the numbers form a program of instructions called a part program designed for a particular work part or job.

1.1 NC and CNC Part Programming: Numerical part programming is the procedure by which the sequence of processing steps to be performed by the machine is planned. It involves the preparation of an input medium used to transmit the processing instructions to the machine tool. There are two types of part programming, namely, manual part programming and computer assisted part programming. Whether it is manual or computer assisted the main objective of part programming is the preparation of input medium.

The part program is converted into a series of machine tool instructions by means of the input medium, which contains the program, and the controller unit which interprets the input medium. The most common input medium is the punched tape. In manual part programming the codes are typed into a typewriter

like machine which produces the punched tape. In computer assisted part programming the tape is prepared directly by the computer using a device called a tape punch. Manual part programming is suitable only for simple cases and it becomes difficult for complex cases. Hence, our further discussion will be limited to computer assisted part programming only.

In computer assisted part programming, the machine instructions are written in English-like statements of NC-programming language, which are processed by a computer to prepare the tape. The computer automatically punches the tape in proper format for the particular machine. The part programmer's responsibility in computer assisted part programming consists of the following two basic steps:

1. Defining work part Geometry

2. Specifying the operation sequence and path

No matter how complicated the work part may appear, it is composed of basic geometric elements. Nearly any component that can be conceived by a designer can be described by points, lines, circles, cylinders, etc. The part programmer has to enumerate the elements out of which the part is composed. Each geometric element must be identified and the dimensions and location of the element explicitly defined.

After defining the work part geometry, the programmer must next construct the path that the cutter will follow to machine the part. This tool path specification involves a detailed step by step sequence of cutter moves. The part programmer can use the various motion commands to direct the tool to machine along the
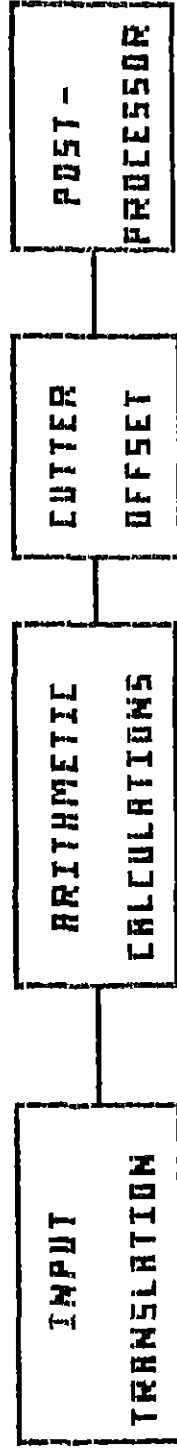
Fig.1 Steps in Computer Assisted Part Programming

work part surface.

The computer's job consists of the following steps:

1. Input translation

2. Arithmetic calculations

3. Cutter offset computation

4. Post processor

Fig. 1 shows the sequence in which these operations are done. Input translation is the translation of program in computer usable form and arithmetic calculation is the calculation required to translate the user specification of geometry into a standard form called canonical form. Once a geometry is specified by the part programmer it is converted into canonical form and for further calculations this is used[2].

In order to machine a particular surface, the tool path(the path of the tool center) must be offset from the work part outline by a distance equal to the radius of the tool. But calculation of this path from the work part geometry is a tedious process. So, in computer assisted part programming, the programmer is required to specify the path assuming the offset to be zero. In this case the work part outline and tool path are identical and so, the specification of tool path becomes simple. It is the job of the computer to calculate the actual tool path from the work part geometry and the radius of the tool.

The part programming languages are usually written for general purpose applications and are not specific to a particular machine. So, a program which can prepare the punched tape, from cutter locations, suitable to a particular machine is required.

Such a program is called a post-processor. It is very specific to a particular machine and hence usually comes along with the particular NC machine.

In general the part programming languages suffer from the following disadvantages:

1. Part programming mistakes

2 Nonoptimal speeds and feeds

3. Wear and tear of punched tapes

In Computer Numerical Control(CNC) these disadvantages are overcome to a large extent. They can sense logical errors and warn the part programmer and optimise speeds and feeds. In these systems the tape is read only once and the machine remembers the instructions for further repeated use. So wear and tear of the tape is avoided.

1.2 Planar Machining Applications: Planar machining is applicable to those work parts, whose surface can be generated by moving a line segment, around a two dimensional contour, at a constant angle in a plane, normal to the contour and passing through the point of intersection of line segment and contour surface. So the path traced by one end of the line will be similar to that traced by the other end but offset by some distance. For these surfaces the input specifications are the two-dimensional contour and the angle of the line segment.

Typical applications are those work pieces that are manufactured by Wire Electro - Discharge Machining(WEDM) or flame cutting. In a WEDM process the electrode wire travels along a path. While travelling, because of the electric discharge it

machining and for cutting those materials which are difficult to cut otherwise, because of their hardness. This process can cut any electrically conducting material irrespective of its hardness[3] This property enables this process to be applicable to machining of punches and dies. An important feature of the process is that both the die and punch can be cut from the same raw work part[$].

Generally, both WEDM and flame cutting require a single pass of cutting to produce the desired, surface unlike milling which requires more than one pass. However, in WEDM, if surface mechanical properties are of importance, more than one pass can be used in some cases. For such applications flame cutting is not applied. Our further discussion will be restricted to WEDM and flame cutting operations requiring single pass of cutting only.

1.3 APT Specifications: In the first section the general methodology of computer assisted part programming was presented. One of the most popular languages used for part programming is Automatically Programmable Tools(APT). Most of the other part programming languages reported in the literature[1], follow APT in many ways. As such APT is a standard part programming language. Since the present work also aims at accomplishing what APT does, it will be worthwhile to see the APT specifications briefly.

There are four types of statements in APT[1,2]. They are

1. Geometry Statements: These define the geometric

elements that comprise the work part.

2. Motion Statements: These statements are used to describe the path taken by cutting tool.

3. Post Processor Statements. These apply to the specific machine tool and control systems. They are used to specify feeds and speeds and to actuate other features of the machine.

4. Auxiliary Statements: These are miscellaneous statements used to identify the part, tool, tolerances and so on.

In order to specify geometry, APT uses the drawing elements such as lines, points, circles, ellipses, cylinders and tabulated cylinders (for specifying arbitrary curves). These elements can be specified in many forms. For example a line can be specified by giving two points through which the line passes or by giving one point and slope In the same way the part programmer has the option of specifying other drawing elements also in any way which will uniquely determine the element. Using these elements the part programmer has to specify the outline of the object.

(eg)   L1 = LINE/P1,P2 (It is a description line   passing through points P1 and P2)

L1 = LINE/P1,LEFT,TANTO,C1 (It is the description of a line passing through point P1 and tangent to circle C1)

Motion Statements: These statements guide the cutter along the outline defined. To accomplish this, check surface, part surface and drive surface are to identified. A typical motion statement will specify a drive surface(the surface that guides the side of the cutter), the position of tool with respect to drive

7

surface(i e, whether the tool is to the right or left of drive surface), direction of motion with respect to previous direction of motion, the check surface(the surface that stops the movement of the tool), the exact position of stopping relative to the check surface namely whether the tool should stop just after touching the check surface or after the tool centre is on the the check surface or after completely clearing the check surface etc. Part surface is used in specifying three dimensional surfaces only.

    (eg)   GOFWD/LI82,PAST,LI83 (commands the cutter to move to forward with LI82 as the drive surface till the cutter clears the check surface LI84)

1.4 NC Programming With Interactive Graphics and PCs: There have been several innovations in NC part programming which increase the productivity of the part programming process. One of the most important of such innovations is part programming using Interactive Graphics. In this method instead of defining the work part geometry using the program, the geometry is drawn on the screen using the drawing editor provided by the system. Using this most of the systems automatically generate the cutter path with minimal additional input from the part programmer. The output resulting from this procedure is a listing of the required APT program or CL File which can be post processed to prepare the punched tape. In addition to the ease of defining geometry, there are many benefits over the APT. With the part drawn on the screen the programmer can overlay the blank on the part to

<center>8</center>

consider the number of passes required, optimal location of work part outline on blank to reduce scrap, to generate alternate method of fixturing etc. Since each step of the process is visually verified by the part programmer, there will not be any ambiguity in his mind regarding the tool path.

Till now most of these softwares required CAD/CAM graphic systems. So their applicability was not wide. With the coming of the PCs having graphic features, such softwares have been developed for PCs also. Some of the softwares accept a part program as input, compile it, generate CL file and graphically show the cutter path to the user for his verification. Others have the full capability of a regular CAD/CAM package, i.e, take in tool geometry graphically as input, and output CL File or APT program.

1.5 Objectives and Scope: The primary objective of this thesis is to design and implement a PC-based part programming software which takes inputs graphically and interactively. The package should generate as output the corresponding CL File. The scope of the package is limited to planar machining applications only. The software is more intelligent than APT in the sense that it can point out many geometric specification errors at the time of specifying itself. Geometry and motion specifications are independent and so the geometry specified can be overlaid anywhere on the blank. When two inserts are to be cut out of a blank, minimisation of scrap could be achieved by careful overlay of geometry on the blank. Though conceptually possible, elaborate

9

provision for specifying auxiliary command has not been provided. The system after generating the cutter location path simulates the motion of the cutter on the blank. The system is applicable to operations of a WEDM and flame cutting machine. Adequate facilities for editing the geometric and other specification have been provided.

CHAPTER 2

PACKAGE SPECIFICATION

2.1 Menu Structure: The menu structure of the package is shown in fig. 2. The geometric input part of the package is highly menu driven and the non-geometric input part works in dialogue mode. There is provision for going to either the previous menu are to the main menu from every menu. The user is asked to specify his option by entering the corresponding number when such a number is provided. When no number is provided he is requested to enter the first character of the option.

2.2 Modules: The package is divided into three modules. The first is the geometry input module, second is the object identification, preliminary cutter path generation and orientation finding module and the third is the cutter path generation and simulation module. Each module is in turn divided into many sub-modules. Fig. 3 shows block diagrams of major modules and submodules. Some of the modules are general utility modules and others are specific.

Because of the memory limitations of GWBASIC the package is physically divided in three files. The details are as follows:

Geometry Input --------- Proact2.bas

File Retrieval ----------- Safile .bas

Rest ------------------ Partloop.bas

Root Menu

1 Specify Geometry
2. Specify Part Progm. Details & Generate CL File
3. Quit/Save & Exit

To CL File Module

Save & Quit

1. Save & Exit
2. Save
3. Quit
4. Root Menu

MAIN MENU

1. Settings
2. Zoom
3. Redraw
4. Erase
5. Draw
6. Root Menu

Settings

1. Screen Lt.
2. Snap
3. Main Menu

Screen Lt

1. Starting Pt
2. Height
3. Width
4. Previous
5. Main Menu

Snap Menu

1. On
2. Off
3. Previous
4. Main Menu

Erase Menu

1. Selected Object
2. Last
3. Main Menu

Erase

1. Entire
2. Partial
3. No Erase

Arc

1. C/SA/EA/R
2. C/SA/EP
3. C/EA/SP
4. C/SP/EP
5. R/SP/EP
6. SP Tgt To
7. EP Tgt To
8. SP&EP Tgt
9. Previous
10. Main Menu

Draw

1. Line
2. Arc
3. Curves
4. Main Menu

First Point

Absolute/Osanp
Previous
Main Menu

Second Point

Absolute/Snap
Previous
Main Menu

ERASE MODULE

```
┌─────────────┐     ┌─────────────┐     ┌─────────────┐     ┌─────────────┐
│   SELECT    │─────│ FIND PORTION│─────│ ERASE FROM  │─────│   UPDATE    │
│   OBJECT    │     │  TO ERASE   │     │   SCREEN    │     │             │
└─────────────┘     └─────────────┘     └─────────────┘     └─────────────┘
```

DRAW MODULE

```
┌─────────────┐     ┌─────────────┐     ┌──────────────────┐     ┌─────────────┐     ┌─────────────┐
│    LINE     │─────│  GET POINT  │─────│   CONVERT TO     │─────│  DRAW ON    │─────│   UPDATE    │
│    ARC      │     │  RELATIVE   │     │  STANDARD FORM   │     │   SCREEN    │     │             │
│   CURVE     │     │             │     │                  │     │             │     │             │
└─────────────┘     └─────────────┘     └──────────────────┘     └─────────────┘     └─────────────┘
```

ZOOM MODULE

```
┌─────────────┐     ┌─────────────┐     ┌─────────────┐
│ GET (FIRST) │─────│    FORM     │─────│   CHANGE    │
│    POINT    │     │  WINDOWS    │     │   LIMITS    │
└─────────────┘     └─────────────┘     └─────────────┘
```
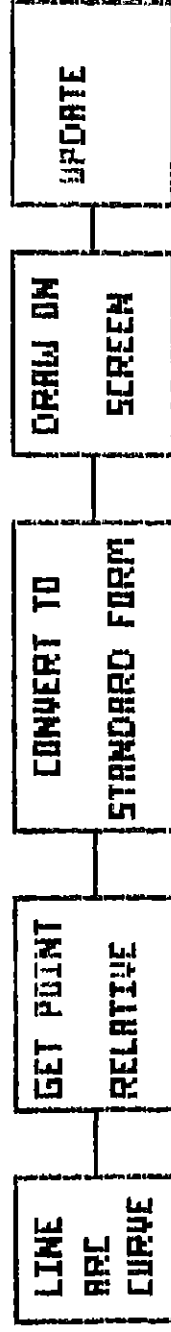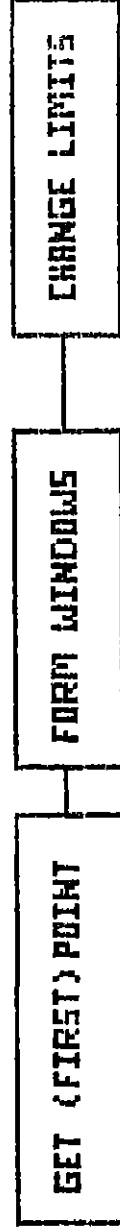
Fig 3.a

SELECT OBJECT

```
┌──────────────┐          ┌──────────────┐
│  GET POINT   │──────────│ FIND OBJECT  │
└──────────────┘          └──────────────┘
```

RELATIVE (OSNAP)

```
┌──────────────┐     ┌──────────────┐     ┌──────────────┐
│  GET POINT   │─────│ FIND OBJECT  │─────│  FIND POINT  │
└──────────────┘     └──────────────┘     └──────────────┘
```

GET POINT

```
┌──────────────────────────┐     ┌──────────────────────┐
│ CURSOR MOVEMENT MODULE   │─────│ VALIDATE TEXT INPUT  │
└──────────────────────────┘     └──────────────────────┘
```

Fig 3.b

CL FILE GENERATION

```
┌─────────────────────────────────────┐
│     CONNECTION IDENTIFICATION        │
└─────────────────────────────────────┘
                  │
            ┌──────────┐
            │  MODIFY   │
            └──────────┘
                  │
┌─────────────────────────────────────┐
│  GENERATE PRELIMINARY CUTTER PATH    │
└─────────────────────────────────────┘
                  │
         ┌──────────────────┐
         │  FIND ORIENTATION │
         └──────────────────┘
                  │
       ┌──────────────────────┐
       │  GENERATE CUTTER PATH │
       └──────────────────────┘
                  │
          ┌──────────────┐
          │  SIMULATION   │
          └──────────────┘
```
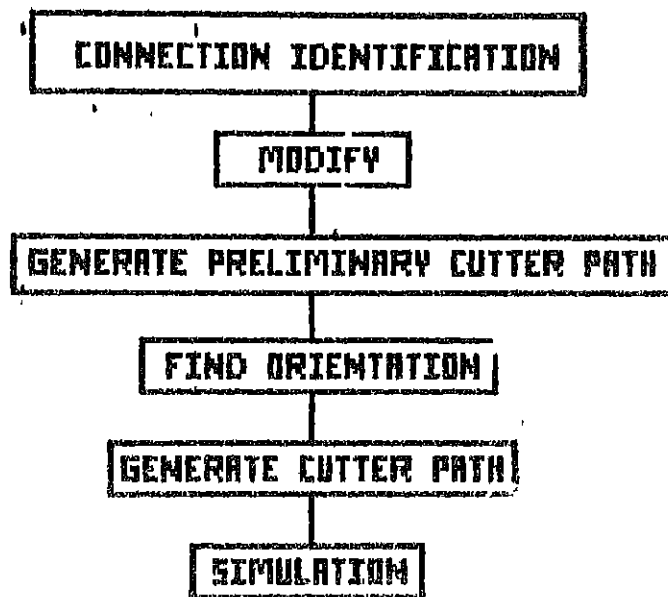
Fig 3·C

2.3 Data Structure: The arrays whose names start with character 'O' are integer arrays and others are single precision real arrays. The list of arrays used by the system and the details of its elements are as follows:

Drawing Elements Details Array

Name : Obdet   Dimension : 22 x 3

Details of Elements:

First dimension refers to the drawing element number. The second refers to a particular attribute of it. An object has 3 attributes. The structure of the array is shown in fig. 4(a).

Line Details Array

Name: Ldet   Dimension : 15 x 4

Details of Elements:

The first dimension refers to the number of a particular line segment in the array. It is this number which is stored in the second and third attribute of obdet when the value of object type attribute(first attribute) is 1. The second dimension refers to the four attributes of the line segment. The structure is shown in fig. 4(b).

Arc Details Array

Name: Ardet   Dimension : 5 x 7

The first dimension refers to the number of a particular arc in this array. It is this number which is stored in the second and third attribute of obdet when the value of the object type attribute(first attribute) is 3. Each arc has seven attributes and they are shown in the structure in fig. 4(c).

14

Obdet

Obj No. \ Attribute	Object Type (1)	Pointer to St. No In Relevant Array (2)	Pointer to End No. In Relevant Array (3)

Fig 4 a

Ldet

Line Seg No. \ Attribute	St. X Co-ordinate (1)	St. Y Co-ordinate (2)	End X Co-ordinate (3)	End Y Co-ordinate (4)

Fig 4 b

Ardet

Arc No. \ ATTRIBUTE	Cent X Co-ord (1)	Cent Y Co-ord (2)	St. X Co-ord (3)	St. Y Co-ord (4)	End X Co-ord (5)	End Y Co-ord (6)	CCW/CW Co-ord (7)

Fig 4 c

Curve Details Array

Name: Cudet   Dimension : 80 x 2

The first dimension refers to the number of the point. If it is the starting point of any curve, this number is stored in the second attribute of the obdet, if the first attribute is **4**. If it is a ending point of any curve, this number is stored in the third attribute of the obdet. Each point has two attributes and they are its two co-ordinates.

Object Details Array

Name : Oloopdet   Dimension : 5 x 3

This array contains details on connected links. The first dimension refers to the number of the connected link. The second dimension has three attributes. The first attribute is a pointer to the number(in obdet) of first drawing element of the link. The second refer to the number of the last drawing element of the link. The value of the third is 1 if it is a closed loop and Ø if it is a open loop.

Details of Clfile Array

Name : Oclfile   Dimension : 5 x 2

This array contains details on individual links in cutter location file. The first dimension refers to the number of the link. The second dimension has two attributes. The first refers to the total number of points in CL File for that object. The second refers to the orientation in which it is stored.(1 for CCW links, -1 for CW links, Ø for open links)

Preliminary Cutter Location Array

Name : Pcldet   Dimension : 2 x 25Ø x 2

This array has the co-ordinates of the points of the preliminary cutter location file. The first dimension refers to the number of the link. The second number refers to the number of a particular point in that link. The third dimension has two attributes which refer to the two co-ordinates of the point.

Final Cutter Location Array

Name : Fcldet   Dimension .5ØØ x 2

This array has the co-ordinates of the points in the final cutter location file. The first dimension refers to the number of the point in the cutter location file. The second has two attributes which refer to the two co-ordinates of the point.

2.4 System Specifications:

Machine Required   . A PC with EGA

RAM   : 64Ø KB

Operating System required: DOS 3.1

Terminal Required   . Color Terminal with 65Ø x 35Ø
                                     resolution

Language   : GWBASIC

It produces the following two files:

        1   Diagram File(.DGM)

        2. CL File (.CLF)

These two files are written only in A: drive. In the present format it runs from A: drive only.

CHAPTER 3

GEOMETRY IMPLEMENTATION

As stated previously, geometry input, auxiliary input and cutter path input are the three most important inputs to a part program In this chapter the implementation of geometric input has been discussed and the next chapter discusses other non-geometric input.

The geometry input module is highly menu driven. It has been designed and implemented with the intention of providing the part programmer, the facility to describe a fairly complex geometry with much ease While specifying geometry one not only needs provision for drawing objects on the screen, but also needs an environment which facilitates the description of geometry. Let us see the environment that has been provided and then we can see the drawing specification part.

3.1 Environment

The screen during the geometry specification of the work part looks as shown in fig. 5. The screen is divided into four areas. The large central area is the drawing area. On this part of the screen the visual area of the drawing is displayed. The right most is the menu area. The right bottom area displays the cursor location and visual area limit. The first row gives the current co-ordinates of the cursor. The second row gives the left bottom co-ordinates of the visual area. The third row gives the right top co-ordinates of the visual area. The left bottom is

the dialogue area It is the portion in which part programmer's inputs are also taken.

<u>3 1 1 Screen Limit Setting.</u> This helps the part programmer to define the visual area of the drawing. Visual area is the portion of the drawing presently displayed on the screen (i.e, the drawing may have a length of 100 units and width of 100 units. But the part programmer may be working in a area of 10 x 10 at a particular time. Then this becomes his necessary visual area). To fix the visual area the part programmer has to give the lower left corner of the visual area and either one of width or height of the screen. Since equal scale for X and Y axis is provided, one fixes the other. The calculation is like this

Height (Given Width) = Width x 270/515 x (5/4)

Width (Given Height) = Height x 515/270 x (4/5)

270 is the height of screen in pixels. 515 is the width. 4/5 is the aspect ratio of the screen.

Magnifying / Reduction of drawings is also done using this facility. Increasing the visual area makes the drawing small. Decreasing the visual area makes the drawing big.

<u>3.1.2 Cursor Resolution Setting.</u> For each pressing of the arrow key the cursor shifts by a power of 10 pixels in that direction. If F1 is pressed the distance moved increases by 10 times and if F2 is pressed it decreases by 10 times. The distance moved depends on screen limits and is calculated like this :

Distance moved for shifting by 1
pixel in the horizontal direction = Width / 515

Distance moved for shifting by 1
pixel in vertical direction   = Height / 270

19

```
┌──────────────────────────────────┬──────────────┐
│                                   │ ROOT MENU    │
│                                   │              │
│                                   │              │
│                                   │              │
│          VISUAL AREA              │    MENU      │
│                                   │    AREA      │
│                                   │              │
│                                   │              │
│                    ┌──────────────┴──────────────┤
│                    │ CURRENT                      │
├────────────────────┤                             │
│  Enter choice:     │ LOWER  0.00    0.00          │
│                    │ UPPER  10.00   6 55          │
└────────────────────┴─────────────────────────────┘
```

Fig. 5

For example, for a width of 1∅ the movement will be a multiple
of ∅.∅19417 in horizontal direction and a multiple of ∅.∅24271 in
vertical direction. Using this it will be very difficult to
position the cursor at any particular point on the visual area
To get over this, Snap Option is provided. When snap is set 'on'
the part programmer is required to specify the resolution of
pointing to a point, in X and Y direction. After specifying this
he can position the cursor at any point whose co-ordinate's
resolutions, are less than the specified one. If the resolution
in X direction is ∅.1 and in Y is ∅.1, then he can position the
cursor at any point whose X co-ordinate is an integral multiple
of ∅.1 and Y co-ordinate is an integral multiple of ∅.1

3 1.3 Zooming: The part programmer may want to have a close look
at a particular portion of the drawing or he may want to draw
something which is very small, in a particular portion of this
drawing. So he may want to make a small portion of the present
visual area as the new visual area. Zooming, is the facility for
doing this. It makes a window of the current visual area as the
new visual area. To increase the visual area he has to change the
screen limit setting once again.

3.1.4 Redraw: Because of the limitations of the terminal, a
drawing element which is being erased may not get completely
erased on the screen though it has been erased internally.
Similarly some points which were given as inputs may be left on
the screen. To remove all drawing elements which are not part of
the drawing the drawing has to be redrawn. Redraw is the
provision for doing that.

21

3.1.5 Erase. It is the most important provision for editing the drawing. The part programmer could have drawn a drawing element by mistake He could also have drawn it intentionally, if having that temporarily gives him some advantage for specifying geometry of some other drawing element In both the cases he will like to erase the drawing element. The part programmer can either erase a complete drawing element or a portion of the drawing element For erasing, first he has to select a drawing element he wants to erase. If only a portion of the drawing element has to be erased he has to specify two points between which he wants the element to be erased.

In case of partial erase the two specified points need not be exactly on the element. If the element is a line, the portion of the line between two points that are closest to the specified points is erased. In the case of an arc, the portion of the arc, that is between two points that are at the same angle from the centre as the specified points is erased. Partial erase is not provided for a curve.

If the points are outside the extreme points of the elements, the extreme point closer to that specified point is assumed. Partial erase is done only in CCW direction in the case of arcs.

3.1.6 Selecting A Drawing Element: A drawing element is selected by keeping the cursor over some point on that drawing element and hitting return. An aperture of small area of 0.01 x 0.01 with the cursor at the centre is assumed. All drawing elements are tested

22

to see whether any part of the drawing element passes through the aperture. Hence, to select a drawing element, at least one point of the element desired to be selected must be within a small area of 0.01×0.01 , with cursor at the centre. If the element is within the aperture, the system changes the colour of that element and asks confirmation from the programmer, for the selection of the element. If more than one element is within the aperture, the colour of the elements within the aperture are changed one by one and confirmation asked, till the part programmer confirms.

3.2 Drawing Options: The package supports three basic drawing elements. They are

        1. Line Segment

        2. Arc of a Circle

        3. Curve

3.2.1 Line Segment: A line segment is determined by its end points. So in order to describe a line segment its end points must be specified. The two points can either be specified in absolute form or relative to some other existing drawing element. Absolute Specifications: It is done either by typing the X and Y co-ordinates with a comma in between or by keeping the cursor over that point and then pressing return.

Relative specifications: These are called Object Snap(OSNAP) options. These increase the flexibility of geometric specifications. Using this, the points can be described by specifying that it is the end point of a line or arc or curve, or

mid point of a line segment or arc, or centre of some arc.

The line segment can also be specified by giving the angle of the desired line segment with the horizontal and its length from the first point. Hence, only the second point can be specified that way. Again, this angle can be specified either in absolute form or in relative form. An angle is specified in absolute form by giving the angle of that line segment(0-360) from the horizontal. It is described in relative form by specifying that the line is normal to some other existing line segment or arc. It can be parallel to some other line segment. The line segment can be tangential to some curve or arc. Fig. 6 shows in how many ways a line segment can be specified.

In some relative specifications there may be ambiguities. The possible ambiguities and the way they are resolved are as follows:

End point of: When the part programmer specifies some point as the end point of a line or arc or curve, then out of the two end points of the selected drawing element, the end point which is nearer to the cursor position used to select the object, is always assumed as the referred end point.

Tangent to: When the line segment is specified as tangent to some arc, and if the first point is outside the selected arc then two tangents are possible. Here, the tangent, whose contact point with the selected arc is nearer to the cursor position used to select the arc, is assumed as the referred tangent. In this case the tangent is drawn upto the contact point from the first point. If the first point is already on the selected arc then part

24

First Point
Absolute
Mid pt of
End pt of
Centre of

X

Second Point
Absolute
Mid pt of
End pt of
Centre of
Normal to
Tangent to
Parallel to

4   7   = 28

Fig. 6 Options for Drawing Line Segment.

programmer is required to specify the length of the tangent.

Normal to: When a line segment is declared as normal to a selected line segment and if the first point is not on the selected line segment, then the new line segment is drawn upto the point of intersection of, the normal to the selected line segment and the line in which the selected line segment is a subset. If the first point is already on the selected line segment then the part programmer is required to specify the length of the desired line segment.

If the line segment is to be normal to some arc, the line segment is drawn from the first point to the first intersection of, the normal to the selected arc and the arc itself. If there is no intersection point at all, an error message is given.

3.2.2 Arc: Arc has been implemented using the circle command in GWBASIC. So, far an arc, the centre, radius, starting angle and ending angle must be determined. It is drawn in CCW direction from starting angle to ending angle. Like a line segment, an arc can be described in many ways. So, a group of attributes from the following, adequate to determine the arc must be specified.

1. Centre
2. Radius
3. Starting Angle
4. Ending Angle
5. Starting Point
6. Ending Point

In the implementation, the combinations that will be very frequently used are provided as options. All the possible

```
Centre,   Radius, St. Angle, End Angle

Centre,   St  Angle, End Point

Centre,   End Angle, St. Point

Centre,   St.Point, End Point

St Point, End Point, Radius

St.Point, Tangential to

End Point Tangential to

St.Point and End Point Tangential to
```

Fig.7 Options for Drawing an arc

combinations of arc drawing are shown in fig. 7.

As in the case of line segments here also relative specification of points is provided. Angles are always got in absolute form. Radius can either be specified in absolute form or by specifying that it is tangent to an arc or line segment

Special cases of arc drawing: Apart from the cases that are described above, there are two special cases. First case is that in which the arc is described as starting or ending, at an end point of a line segment or arc and tangential to it. In the second case, the desired arc is described as ending and staring at the end pt of line or arc, and tangential to both. In the first case the part programmer has to specify the radius and the subtended angle of the arc but in the second case that is not necessary.

When centre, starting point and ending point are specified by the part programmer there is a possibility of inconsistency because, the radius calculated from starting point may be different from that calculated from ending point. Hence the radius is always calculated from the first point and the second point specification is used for calculating the ending angle only.

3.2.3 Curves: This is a provision for drawing an arbitrary curve. This is mainly a provision for curve fitting than for curve design, since part programming takes place only after the completion of design. The part programmer has to specify the set of points through which, he wants the curve to pass. The implementation uses a parametric tension cubic spline to generate

the curve This algorithm ensures limit continuity and slope continuity. The part programmer, in addition, has to specify the starting and ending slope. Here also, the slope can be either absolute(in the form of angle, $0-360$) or relative. The options of relative form specify, that, starting/ending slope is, same as that of some line segment or slope between the first two specified points in case of starting slope and the slope between last two points in case of ending slope.

This algorithm calculates the slope at an intermediate point as the quotient of, the division of the slope of the imaginary straight line passing through its two adjacent points by a factor called stiffness factor. This stiffness factor determines how close should the curve be to the line segments joining the specified points. If the stiffness factor is high then the curve between any two specified points is almost a straight line. If the stiffness factor is very low the curve deviates from the line segments joining the specified points to a great extent. For most cases a stiffness in the range of $(0.8 - 2.5)$ will be suitable. At low stiffness kinks are also formed in between specified points. So, the part programmer has to specify an initial stiffness, and once the curve is drawn, he can keep on changing the stiffness till the curve he intended is produced. The stiffness is always a positive real number.

The algorithm requires that a minimum of three points be specified. In the present implementation, due to memory limitations only a maximum of eight points have to be specified. In between, the curve is approximated by a series of small

29

straight lines. The more the number of straight lines between adjacent specified points, the smoother the curve is. But, again it requires more memory. Because of that limitation, the number of straight lines between adjacent points is limited to 5

It may be noted that the drawing elements can be specified in many ways. However, the package converts them into a standard form and stores them. This helps in further calculations and searches for identification of drawing elements. In the case of a line segment the starting point co-ordinates and ending point co-ordinates are stored. In arcs, centre co- ordinates, starting co-ordinates, ending co-ordinates are stored. In curves the co-ordinates of various points through which the curve passes ,is stored.

The maximum number of drawing elements that can be drawn is limited to 22. It includes a maximum of 15 line segments, 5 arcs and 2 curves. If any attempt is made to create more, the system gives an error message. When a drawing element is partially erased, it is broken into two drawing elements of the same type. Hence there is a net addition of one drawing element. If it so happens that the limit is already reached, at the time of attempt of partial erase of a drawing element, partial erase is refused. Under these circumstances, at least one object should be completely erased, before proceeding further with the creation of a new drawing element.

3.3 Saving and Retrieving: The part programmer can save the drawing at any point of time. They are saved in the floppy diskette. In the same way, the user can retrieve a drawing for

further editing. The drawings are stored with a .DGM extension. There is a provision for listing all the existing drawing files in the directory of the floppy diskette.

CHAPTER 4

NON - GEOMETRIC MODULES IMPLEMENTATION

In this chapter we shall discuss the rest of the implementations other than the geometry. There are two major non-geometry modules, namely object identification and cutter path generation.

4.1 Object Identification: The part programmer while describing the work part geometry can follow any order which is most convenient to him. Hence, after the completion of drawing specification, the adjacent elements in the drawing elements array need not be connected with one another. But for generating the cutter location file, the system should know about the interconnection of drawing elements. The purpose of this module is to identify the interconnection of drawing elements.

In identifying the interconnections, the two following assumptions are made by the system:

1.   A point can be at the most common for a maximum of two drawing elements. That common point must also be an end point in both the drawing elements.

2.   If the distance between the end points of two drawing elements is less than a particular distance called the aperture distance, then the two drawing elements are assumed to be interconnected, if both the end points have not been identified as common with some other point before.

32

This module identifies all the interconnection between the drawing elements that have been specified Some drawing elements may be connected in such a way that they form a closed link. Others may not form a closed link and will be open. Once it is identified that two drawing elements are connected with one another the drawing elements array and the relevant line/arc/ curve arrays are modified in such a way that the drawing elements follow one another in the same way as they appear on the screen A linkage is assumed to end if no common point could be found between either the starting drawing element of the link and one of the unconnected drawing elements or ending point of the link and one of the unconnected drawing elements. Once, a linkage ends a test is made to see whether the starting drawing element has any common point with the ending drawing element. If there is such an interconnection the link is a closed link otherwise it is open.

', Though all the linkages can be identified, only for two linkages, the CL File can be generated because of memory limitation. At the end of identification of linkages, the system displays the total number of linkages(i.e, complete objects) found and also states how many of them are closed and how many of them are open. If this is not what the part programmer intended, he has to go back to the geometry input part and correct them accordingly.

In view of the two assumptions stated earlier, if three or more drawing elements have a common point, the system will identify two of them to have a common point. The rest of the

elements will not find a place in the same link or object. Hence any work part in which three drawing elements have a common point cannot be processed.

4.2 CL File Generation: For generation of CL File the part programmer has to give inputs for auxiliary commands and motion commands. The information required for generation of cutter path is very much less than APT because, here it is assumed that, the cutter should travel around the work part outline with the required offset. In the case of closed objects the offset will be positive, negative or zero. But for open objects it will always be zero. Hence, the geometric inputs determine the cutter motion to a large extent and need not be specified by the user.

For example, if the work part shown in fig. 8 is to be cut the part programmer, has to specify the drawing outline. There is no need for explicitly specifying the cutter path, other than specifying the direction of progression(CCW/CW) and starting point. But in APT to produce the same effect, first all the lines(not line segments) have to be defined. Then the tool should be guided in the following way.

Let intersection of L4 and L3 be the starting point.

1. Goto starting point.

2. Go along L4 with the cutter to the right of L4, till the cutter clears L1

3. Go along L1 till the cutter clears L6.

4. Go along L6 till the cutter clears L2.

5. Go along L2 till the cutter touches L5.

6. Go along L5 till the cutter clears L4.

7. Go along L4 till the cutter clears L1.

8. Goto starting point.

The example shows the amount of input required to specify cutter path in APT. In the present system, strictly speaking, the only ~~motion command is the~~ direction of progression and in the case of open objects, that too is not required. The block diagram of the process of generation of CL File is shown in fig. 3.

4.2.1 Preliminary Cutter Location Generation: This process produces the cutter locations required for each object assuming the offset to be equal to zero. It is mainly a process of approximating the work part geometry using straight lines and finding the orientation(CCW/CW) in which the closed objects are stored. Since the curves are already series of line segments, one needs to approximate only the arcs. The approximation depends on the tolerance. For this the part programmer has to provide the tolerance. Higher the width of the tolerance band, fewer line segments will be used to approximate and poorer will be the approximation. Fig. 9 shows how approximation is carried out.

4.2.2 Finding the Orientation of Objects: Even though all the objects have been identified ,still the system does not know in what order the closed objects are stored(i.e,CW/CCW). But it will be needed because the cutter location file should have points in the order specified by the part programmer. In order to accomplish that the order in which they are stored must be found out. For that the following algorithm is used:
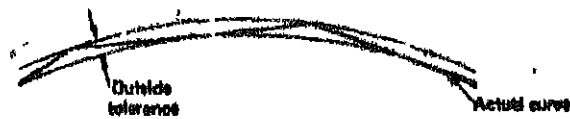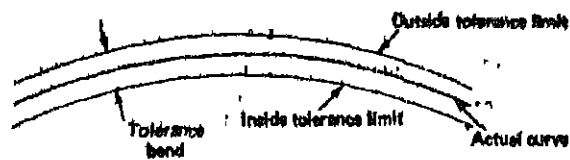
35

L1 L2 L3 L4 L5 L6

Fig 8
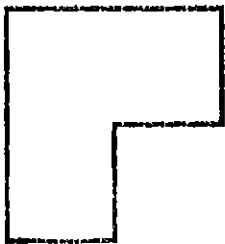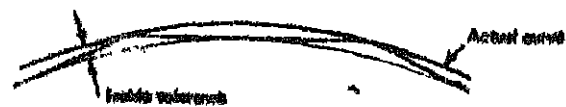
(a)

(b)

(c)

Approximation of a curved path in NC by a series of straight-line segments. The accuracy of the approximation is controlled by the "tolerance" between the actual curve and the maximum deviation of the straight-line segments. In (a), the tolerance is defined on the inside of the curve. It is also possible to define the tolerance on the outside of the curve, as in (b). Finally, the tolerance can be specified on both inside and outside, as shown in (c).

Step   1   Find the left extreme point(point having minimum X
co-ordinate) of the preliminary cutter location of
the given object. Let its co-ordinates be (x_L, y_L)

Step   2   Case 1: If $(y_{L-1} <= y_L <= y_{L+1})$ then direction

L -> L-1 is CCW

Case 2: If $(y_{L-1} >= y_L >= y_{L+1})$ then direction

L -> L+1 is CCW

Case 3: If $(y_L >= y_{L-1}$ and $y_L >= y_{L+1})$

or If $(y_L <= y_{L-1}$ and $y_L <= y_{L+1})$ then

let A_{L-1} be angle of line LL-1

let A_{L+1} be angle of line LL+1

if $A_{L-1} <= A_{L+1}$ then direction of L -> L-1

is CCW else direction of L -> L+1 is CCW

4.2.3 Cutter Path Generation: To determine and generate the
cutter path the following details are provided by the part
programmer in a dialogue mode.

1.   If it is a closed object he specifies, whether the
work part is a hole/insert/punch and die.

2.   If it is an insert then the part programmer
specifies whether pilot hole/band saw entry is
required.

3.   If pilot hole entry then specifies point
corresponding to pilot hole.

4.   If band saw entry then he specifies the starting
point and point on the work part outline which

should be cut first.

5   If open link, starting and ending approach points, if needed, are specified by the part programmer.

6   Direction of Progression(CCW/CW) in case of closed objects.

7.   Diameter of Wire/Flame in case of closed objects is also specified

Whether the work part is a hole/insert/punch and die determine the offset. For hole, the offset is the radius of cutter but inside the closed link. For insert the offset is radius of cutter but outside the closed link. For punch and die the offset is zero. Other information requirements are straight forward

Now, that all the details are available the system can generate the cutter path. The cutter path is nothing but the preliminary cutter path, offset by the necessary value in the correct direction. The cutter path is calculated using the following algorithm.

Step 1   Find the angle from first point to second point in preliminary cutter location of the object. If the direction of progression is CCW then direction = 1 else direction = -1
First_normal = angle - direction x 9Ø
Second point is the current point

Step 2   Take the next point
Find angle from previous point to current point
normal = angle - direction x 9Ø

Step 3   Bisector = (First_normal + angle)/2

Step 4   Angular Gap = absolute of (First_normal - normal)

Step 5   Offset = Radius of Cutter/sin(angular gap)

Step 6   Offset = -1 x offset if hole

Step 7   Find a point at an angle of bisector and length of
          offset   from current point.   This is the CL   point
          corresponding to the current point

Step 8   First_normal = normal
          current point = next point

Step 9   Goto step 2 till all the points are finished.

If specified direction of progression and orientation of object
in preliminary cutter path are same, then the preliminary cutter
points are processed in the same way as they are stored in the
preliminary cutter path. If they are opposite, these are
processed in the opposite direction. In this way, all the objects
are processed. In the case of band saw entry the part programmer-
specified starting point becomes the first point in the CL File.

     In band saw entry, the cutter should come back to the
starting point. Therefore, the last point is same as the starting
point and the last but one point is same as the point of work
part outline at which the cutting was started. In the same way,
in the case of pilot hole entry, the first and last point is the
point corresponding to pilot hole, as specified by the part
programmer.

4.2.4 Blank Positioning: Since the system works in a interactive
and graphic mode, the system can handle work part overlaying on
the blank also. So the chance of mis-positioning the blank and
tool which is possible in APT, is avoided here. Moreover, when
two inserts are to be cut from the same blank, the two can be
overlaid at any position on the blank as desired by the part
programmer. By judiciously overlaying the outlines on the blank

scrap reduction could also be achieved.

For the purpose of overlaying the outline anywhere in the
blank, the system first draws the outline tentatively at some
point. Then the part programmer can specify translation and
rotation about a point in order to transform the outline to the
desired position. If any part of the outline of the closed path
goes outside, then the system gives the necessary error message
and the outline can be transformed once again. If a portion of
open object goes out of blank, it is assumed that it is
intentional and the point of intersection of blank boundary and
outline is taken as the approach point.

4.2.5 Changing Specifications and Saving: It is the APT practice
to give liberal inside and outside tolerances during preliminary
runs and generate cutter location points. Further the CL points
are viewed on   the screen to test them for correctness. If the
points are alright then the tolerances are tightened, because
tight tolerance increases the number of cutter location points
and takes more time for its generation. Following the same
practice here, provision for changing the specifications is
provided. Once the part programmer is satisfied he can save the
CL File. It is stored with the extension .CLF

4.2.6 Simulation: The generated cutter is saved the system
provides a limited simulation of the process also. The blank
position is displayed. If pilot holes are to be put the same is
specified in terms of its location. After that the simulator
shows the way the cutter travels along the blank, producing the
required work part.

CHAPTER 5

TEST EXAMPLES

In this chapter four typical examples are presented to demonstrate the capabilities of the proposed part programming system. For all the examples, the work part outline and cutter path are shown. Example No. 3 has all the features which can be handled by the part programmer. Therefore, this example has been explained in greater depth. The figures for all the examples show the blank length, blank width and cutter diameter. The output, apart from cutter path, also shows the resulting work part.

Procedure For Example 3:

Step 1   Set Snap "on".

Step 2   Select Option 5(For drawing) from Main Menu after setting Snap

Step 3   Specify L1, (by selecting line option in Draw Menu) a vertical line of length 1 unit

Step 4   Draw A1,(by selecting Arc option) with its Starting point tangential to L1
          Radius = -1. Angle = 120

Step 4   Draw L2 specifying first point as end point of A1. It is tangential to A1 and 1 unit long.

Step 5   Draw L4 parallel to L1 and 6 units away.

Step 6   Draw A2, with its Ending point tangential to L4

              Radius = 1. Angle 120

Step 7   Draw L3, specifying first point as the end point

              of A2. It is tangential to A2 and 1 unit long

Step 8   Draw L5 between the end points of L3 and L2

Step 9   Draw A3 with its Starting point tangential to L4.

              Radius = -1. Angle = 90

Step 10   Draw A4 with its Ending point tangential to L1

              Radius = -1. Angle = 90

Step 11   Draw C2 with Ending points of A3 and A4 as

              extreme   points.   Let   the   Ending

              point   of   A4   be the Starting point of curve and

              Ending   point   of   A3   be   the   Ending

              point   of   curve   Then   starting and ending

              angle will be 180 degrees. End specification of

              points by hitting a return without moving the

              cursor

Step 12   Now the figure is complete. Go to Root Menu.

Step 13   Save the outline.

Step 14   Come to Root Menu back and select option(2) for

              generating CL File.

Now, the system gives a brief summary of the drawing. It says the

total number of objects 1: closed 1: open 0

Let the following be the specifications for the present work

part.   The work part is a hole,the required direction of

progression is CCW and the wire diameter is 0.5 mm. The inner

43

tolerance is 0.001 and outer tolerance is 0 001. The blank length
is 10 and width is 10.

The dialogue between the part programmer and the system will be
as follows:

SYSTEM   :Objects :1 Closed 1: Open 1
             Want to Continue/Go back and Edit(E/C)

PART   :C(no Return)
PROGRAMMER

SYS   :Unit of the drawing mm/cm (m/c)

PP   :m

SYS   :Enter Outer Tolerance:

PP   :0.001

SYS   :Enter Inner Tolerance:

PP   .0.001

SYS   :Enter Length of Blank:

PP   :10

SYS   :Enter Width of Blank:

PP   :10

SYS   :Enter the position of Blank:

PP   :(can move the cursor or enter the position)1,1

SYS   :The Object is Hole/Insert/Punch and die (H/I/P)

PP   :h

SYS   :(Draws the object approximated on the screen)
             This is only Temporary Placing Press a Key to
             Continue.

PP   :(strikes a Key)

SYS   :Transforms the object into red colour.

44

Enter Translation of Red Object
(First Point/No Trans/No Cut)

PP   :(Enters First Point)

SYS   :(Second Point/Refix First Point/No cut)

PP   :(Enters Second Point)

SYS   :Data For Rotation
           (Point/No Rotation/no Cut)

PP   :n(for no rotation)

SYS   :(Transforms the object to new location)
           Enter the Point Corresponding to Pilot Hole

PP   :(Enters a Point)

SYS   :Enter Wire Diameter:

PP   :0.5

SYS   :Enter Pilot Hole Diameter:

PP   :1

SYS   :(Draws the original object in blue and draws the
           cutter path in pink)
           (Save/Quit/Retry)? (S/Q/R)

PP   :S

SYS   :Enter C-L File Name:

PP   :(Enters a Name)

SYS   :End Session/Create New Drawing/Simulate (E/C/S)

PP   :S

SYS   :Enter surface roughness permissible

PP   :1

SYS   :Enter the thickness of blank

PP   :4

SYS   :(Pilot Hole Location ***,****)
           Press a Key For Pilot Hole Entry
PP   :(Presses a Key)

Figure 1. The part outline made of arcs and line segments.
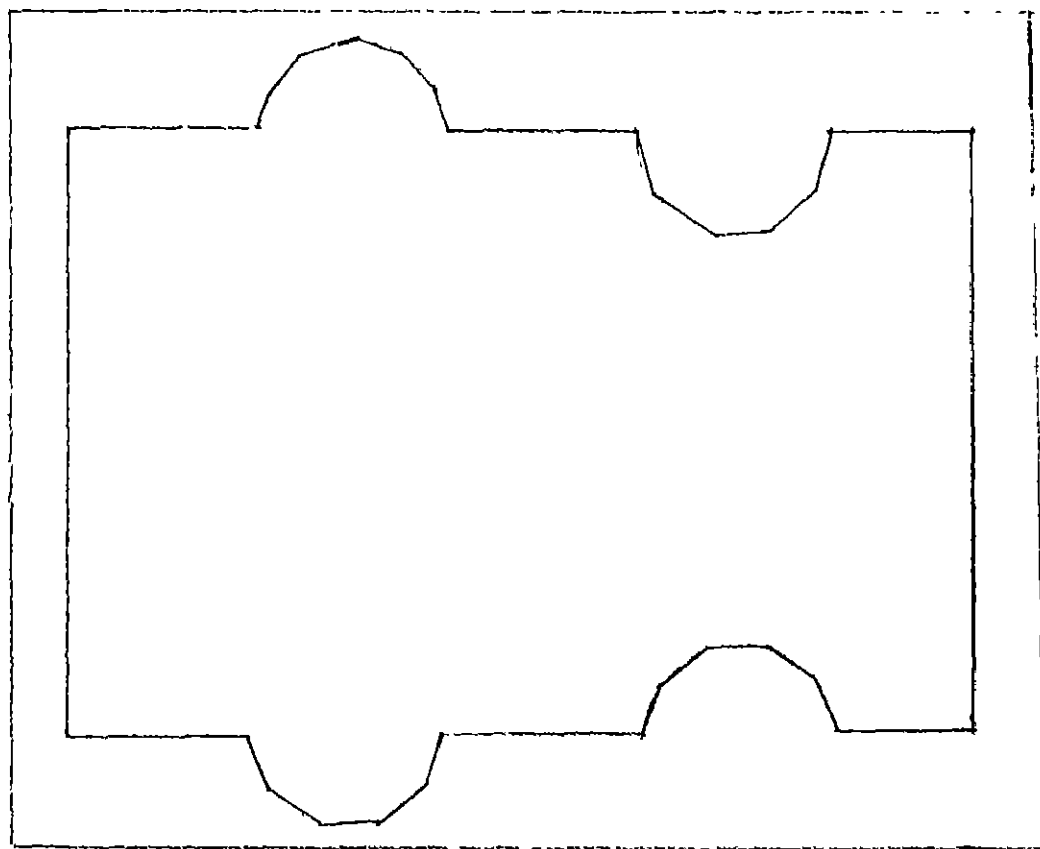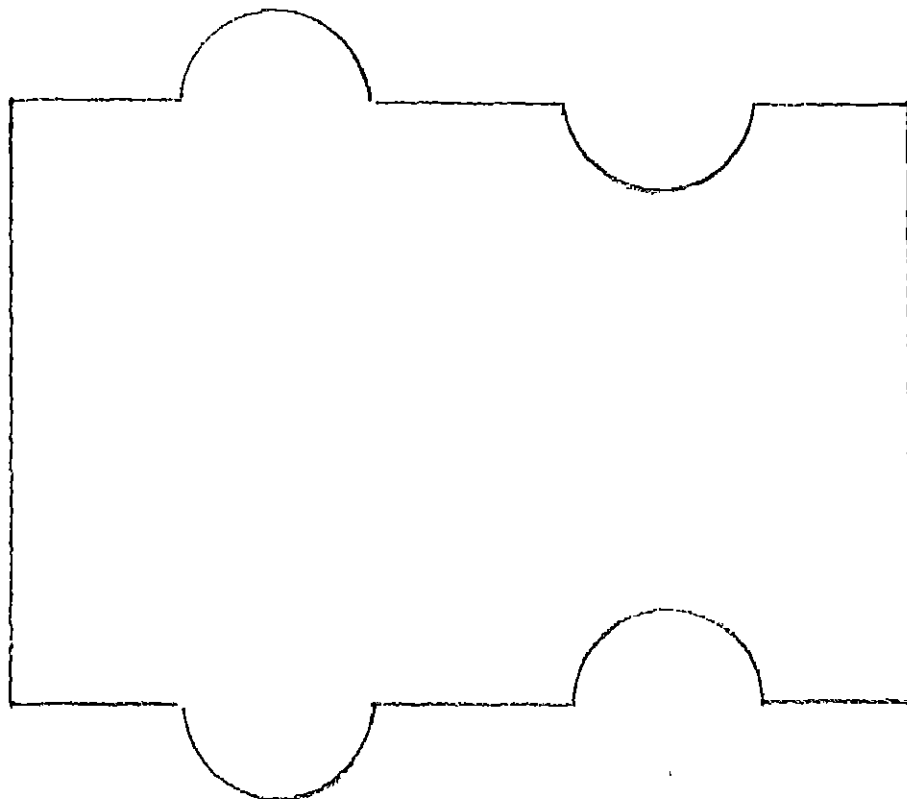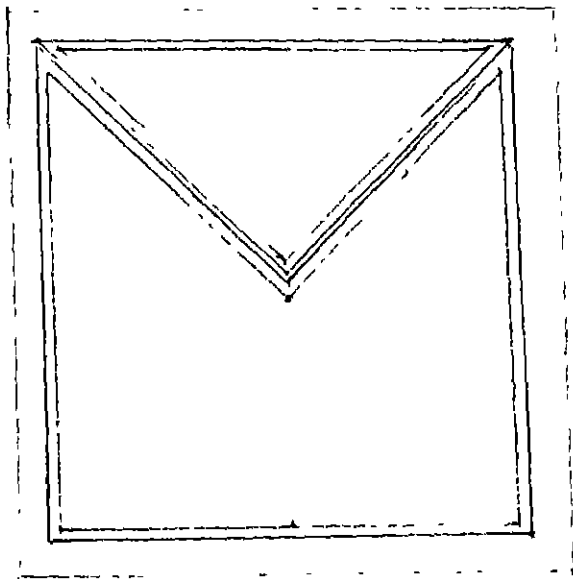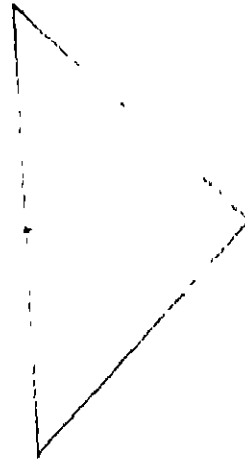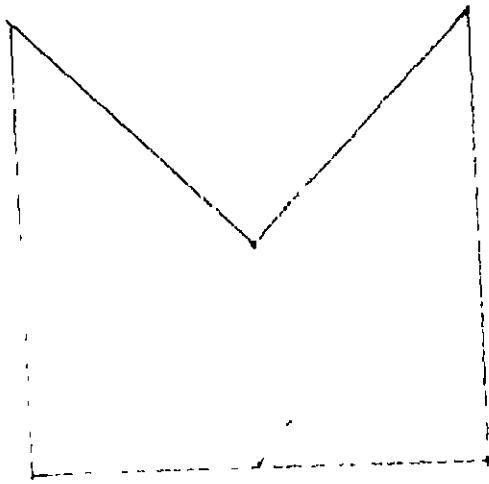
Operations: Punch and die

Cutte r dia o.5
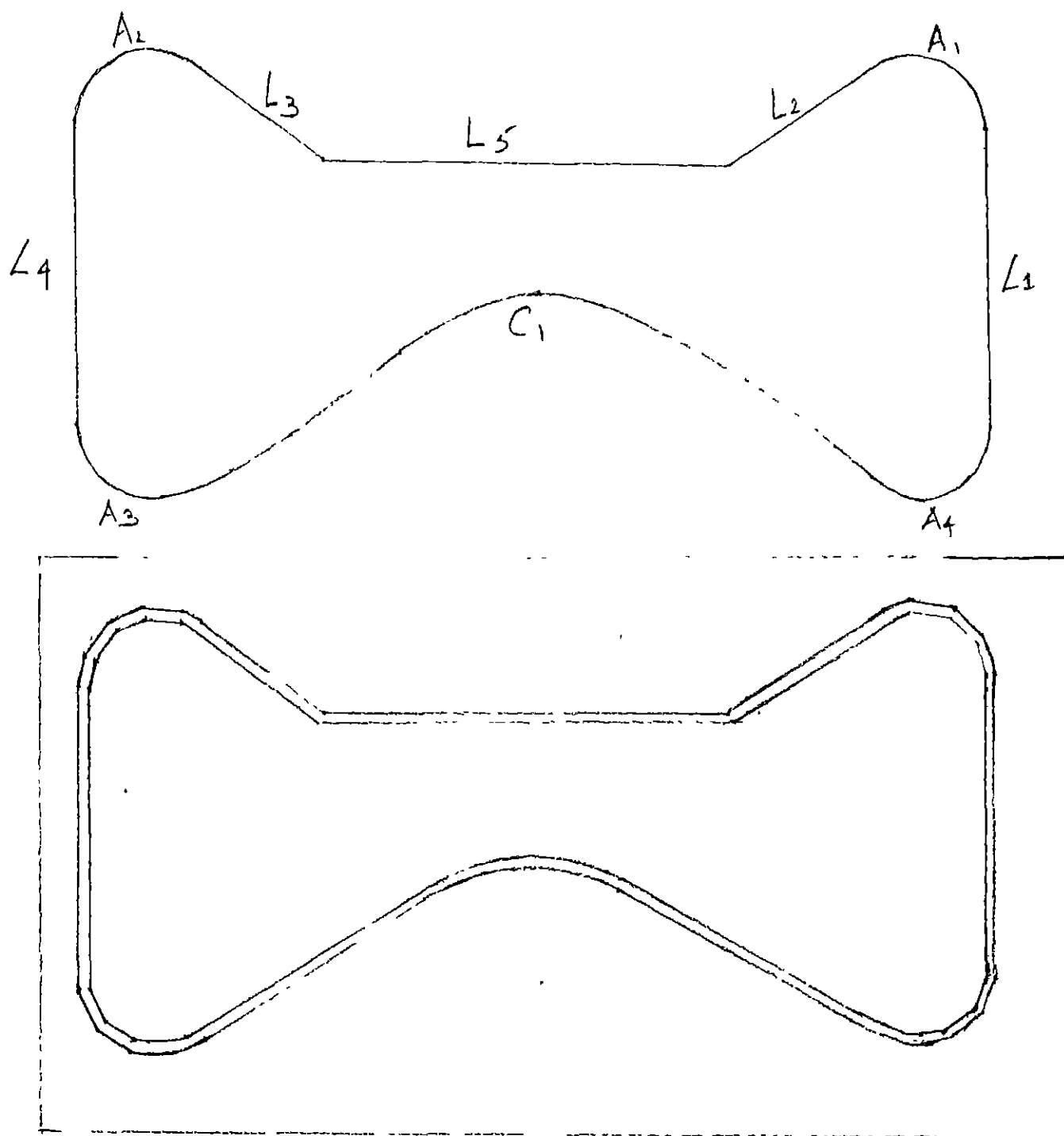
 Blank 15xlo

Offset = 0

Example 2.

An example illustrating scrap reduction. The two outlines are drawn at
diffe rent places. Both are inserts. By suitably overlaying them
on the blank scrap reduction is acheived.

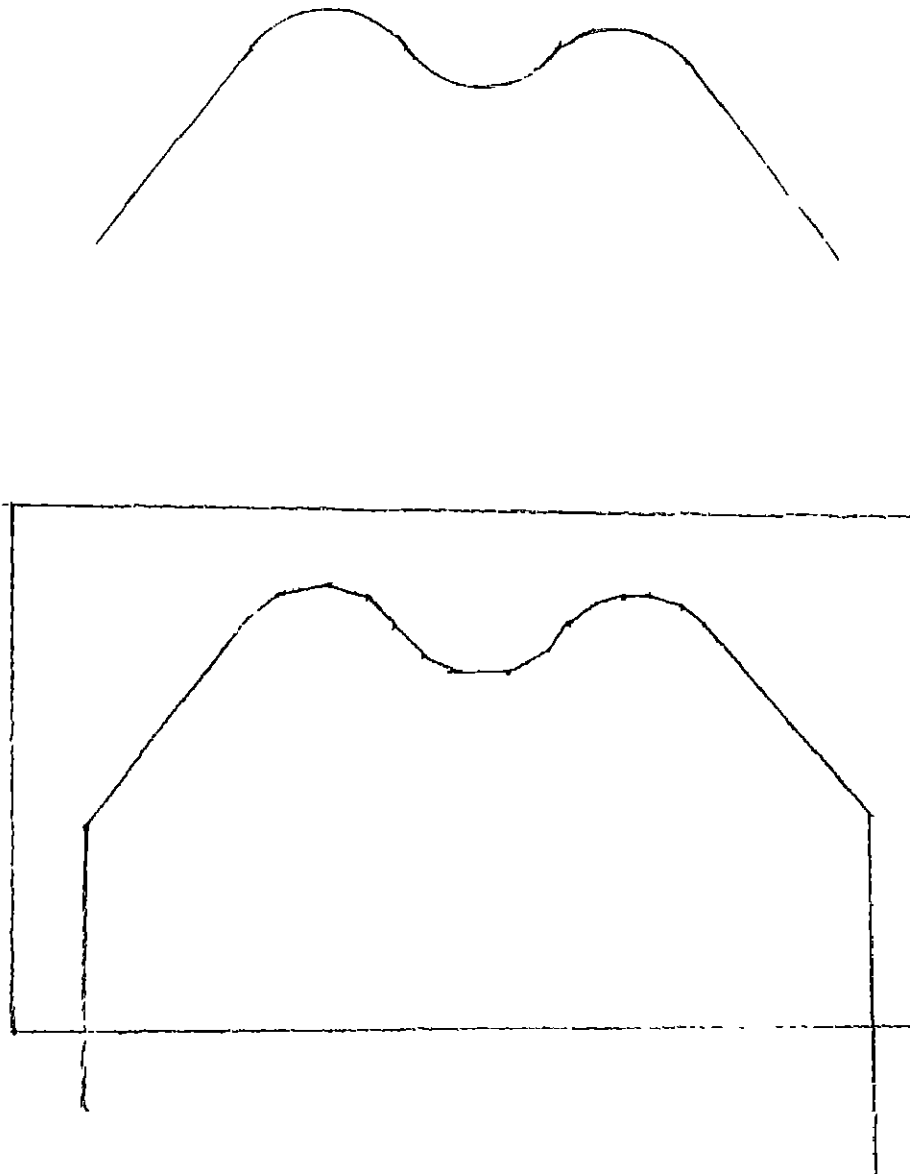Cutter dia 0.5, Offset outside, Blank 2.5x2.5

Example 3.   A hole

Cutter dia 1,   Offset inside

48

Example 4.

Open link: It is drawn by making archs tangential to lines
            and other arcs. It shows the cutte r movement from
            approach points.

CHAPTER 6

CONCLUSION

6.1 Technical Summary: The part programming system has been
tested successfully. It is found to be suitable for describing
fairly complex geometries. It has elaborate features for
specifying arc, line and curve. The generation of cutter location
points is accurate. Since, a post-processor has not been written
the cutting action has been simulated. The part programming
system assumes that circular movement is not available with the
CNC machine concerned. This increases the applicability of the
system. The least count of the machine has not been taken into
consideration while generating the CL File.

Most of the limitations of this software are due to the
memory limitations of GWBASIC. It allows a maximum of 22 drawing
elements. Maximum number of lines is 15, arcs is 5 and curves is
2. Maximum of 8 points can only be specified for a curve. Only 5
line segments are drawn in between any two specified points.

Arcs are drawn and erased only in CCW direction. Curve
cannot be erased partially. The part programming system can
handle a maximum of two objects only. While approximating an arc,
the minimum length of the straight line segment is assumed to be
Ø.1. In all a cutter location file can have a maximum of 5ØØ
points. The corner radius in internal corners is dependent on
cutter diameter. In the present format the part programming

system cannot be run from the hard disk.

6.2 Suggestions for further work: To overcome the memory limitations of GWBASIC, the whole system could be rewritten in Turbo pascal. This would speed up the response time also. Provision for adding a mouse to move the cursor could be made. Since the part programming system uses high resolution screen, the diagrams cannot be printed using print screen command. So an assembly program to accomplish this could be written. Now the system does not have provision for elaborate inclusion of auxiliary commands. This could also be incorporated. Moreover, a data base of EDM and flame cutting parameters could be developed. This will help in the optimal selection of the concerned parameters for a user specified material removal rate and surface finish. These selected parameters could be added as auxiliary commands to the CL file.

An interface to convert the cutter location file into punch tape should be written. Once this is done the system can be put into actual use. The drawing capabilities can be increased by providing more geometric figures like arc of an ellipse, parabola, hyperbola etc.

REFERENCES

1.   Groover, Mikell,P. and Zimmers, Emory,W.,

     'Computer Aided Design/Computer Aided Manufacturing'

     Prentice-Hall of India, New Delhi, 1986.

2.   Kral, Irvin, H.,

     'APT Programming'

     Prentice-Hall International.

3.   Pandey and Shan,

     'Modern Machining Processes'

     Tata-McGraw Hill Publishing Company Ltd., New Delhi,1985.

4.   Microsoft Corporation,

     'GWBASIC Interpreter Manual'

5.   Illinois Institute of Technology Research Institute,

     'APT Part Programming'

     McGraw-Hill Book Company, New York,1967

MEP-1900-M-KRI-DEC